BENCHTEST

THE LYNX

We've been expecting this machine for a while. But is it as slick a cat as it's cracked up to be? Maggie Burton finds out.

Everybody must have seen the adverts by now. 'How to increase the size of your memory.' 'Expandable to 192k.' The Lynx, expected in November, is now just about to go on sale. And having been proclaimed in advance for several months, is it what it claims to be?

Camputers was formed in winter 1981 under the name Camtronic Circuits. Brought together by the Lynx, the company comprises nine members and its sister company, GW Design, about 16. The Lynx itself is subcontracted and the original machines were designed and put together by Camputers and GW.

The machine offers both colour and high resolution graphics. It also offers a 'grow with the user' type of quality that many machines cannot boast. It is theoretically possible to set it up as a small-scale business micro.

Hardware

To look at, the Lynx is very pleasing. It's cased in arty dove-grey injection-moulded ABS and is that rather familiar home micro 'wedge shape' seen from the side. It's about the same size as the VIC-20 but a bit more angular, like an Acorn Atom. The dimensions are 34.5cm wide, 21.5cm from front to back and 6.5cm at the highest point.

The casing is substantially built although there are one or two areas where it 'gives', most notably on either side of the space bar and above the ports at the back. These areas could have done with a little more reinforcement but it could take quite some bashing nevertheless.

A metal-cased PSU plugs in through a DIN socket at the back on the left. The PSU itself is roughly as heavy as the Lynx, weighing just over three pounds on my bathroom scales (the Lynx weighs just under four pounds). The lead between it and the computer is a little short, but the lead from PSU to mains is much longer. The PSU becomes almost painfully hot to the touch after being switched on for more than an hour or so, and at one stage I even warmed my feet on it while computing! The rivets holding it together at the bottom really do get too hot to touch. This could easily have been rectified by cutting ventilation slots in the PSU case and electrically insulating the whole thing from within.

Amusingly, the Lynx comes complete with an optional set of stick-on rubber feet for the PSU (I think they'd possibly melt if you stuck them on to the rivets). Otherwise, they stick like limpets; I stuck some onto a Sirius and I think they're there to stay.

The I/O connectors are nice and solid although you can see into the machine above them. From left to right (with the front of the machine facing you) there's power, expansion bus, cassette, RS232, light pen and composite video, RGB and, on the far right, the TV socket.

The keyboard comes in the same grey colour with white lettering on all character keys and red on the control ones. There is a total of 57 keys, 12 being devoted to control. These are: ESC, Control, up and down arrows, Shift Lock and Shift on the left; Break, Delete, left and right arrows, Shift and Return on the left.

Escape is used to exit a program in the middle of execution. The Break key is used (to quote the manual) 'within certain specialised programs' but it does not go into any further explanation. Apparently this key will be used in future applications. There is no reset key, which proved to be a nuisance — to get out of an infinite loop or to recover from a total crash, you have to unplug the PSU.

I noticed with some amusement that the Return key is labelled 'Control' even though there's a Control key, too. This is almost certainly a mistake as other Lynxes I've seen don't have this peculiarity. The big disadvantage with the Return key is that it's parked right next to the Shift key. If you're typing away enthusiastically it's very easy and extremely exasperating to hit Return instead of Shift — which, of course, means you have to retype the whole line. This would have been better placed where the Delete and arrow keys are and would thus prevent a lot of lost time.

Typing on the Lynx keyboard is quite a pleasant exercise. It has a crisp feel to it and responds well. It's endowed with auto-repeat, although this is a little slower than is necessary, and is fully debounced.

Inside

The machine reviewed here is the standard 48k machine; 92k, 128k and 192k versions are available and smaller machines can be fully upgraded internally. An upgrade to either 128 or 192k will increase the screen resolution from 248 x 256 to 248 x 512 and give an 80-column display as well but you'd have to buy a monitor for this.

In spite of the fact that I'm hardly either a stickler for tidiness or a hardware designer, I did feel that the design of the PCB was a little inelegant. This is probably made worse by the fact that the Lynx uses no ULAs and therefore has to employ a larger number of smaller chips to do the job. RAM is not grouped in any apparent logical order on the board — 16k of it is in front of the keyboard plug, 16k is haphazardly arranged on the right of the Basic interpreter chips and 16k comes on a cute little plug-in board which goes just in front of the TV socket.

The 4MHz Z80 sits between the keyboard plug and the Basic interpreter. The Basic interpreter comes on two EPROMS (socketed, of course) which will probably turn into ROMs before too long. A third socket is provided, presumably for the insertion of dedicated ROMs/EPROMs.

To the far left at the back of the Lynx sits a fairly large speaker (through which the Lynx beeps disconsolately when switched on), cushioned by a circular piece of plastic foam.

RAM expansions almost certainly plug in where the aforementioned memory board does, but there is a plug just in front of the parallel expansion bus which is probably also designed for extras. In fact, as there's a good bit of fresh air above that plug, you could conceivably fit in a substantial cartridge or circuit board — an indication of room being made for the future.

All in all, it's a solidly put together but really somewhat dated piece of computer engineering. All soldering is precise, components are firmly fitted and wires properly trimmed. Compared to, say, the Spectrum, it's a rather clumsy outfit when you consider the fact that the two machines have similar capabilities, but size and appearance really aren't everything.

Display

When you first switch on, the Lynx prints its logo at the top left of the screen (complete with the little pawprint) in white on a black screen. Underneath this it puts the prompt 'Ready!' and a 'greater than' sign followed by a flashing block cursor.

The display is quick to appear and very crisp. Initial tuning in is very simple and, once set, it rarely drifts off frequency. The only problem here is that the display is rather too sensitive to the position of the TV leads. Jerking this or loosening it slightly (as on occasions when the TV gets knocked) can cause the screen to waver. The colours are very vivid and crisp and, once you have the machine tuned right, they never drift. Full upper and lower case lettering is provided directly from the keyboard.

Photography by Karin Craddock

Performance varies from TV to TV. I found the Lynx wouldn't produce colour at all on a nine-year-old Ferguson (but then neither would the Spectrum) but it worked well with a Sony Trinitron and even better with a small Hitachi portable. If in any doubt — for instance, if your TV is ancient or a little-known model — some consultation with a dealer would be wise.

Lynx Basic

David Jansons, who designed Lynx Basic, had a lot of experience of Microsoft and TRS-80 Basic. 'I don't like them', he said. His aim in putting together a highly non-standard Basic was to make it 'as structured as possible'. Considering he put the original interpreter together in a little over four months, it's quite an achievement, though it has been regularly amended since the machine was announced in September.

It includes some very odd qualities indeed, the reasoning behind which seems a little strange. First and foremost, it won't allow multi-statement lines. Now, quite a few older machines are the same, and Jansons explained he did this to improve readability of listings. But the alternative in the area of code-cramming is to use line numbers with a *decimal point!* This means you can have a huge number of lines in a program — four figures after the point are allowed — and this is far more than you could ever need.

Not only is this facility negated by the Lynx's RENUM command (which will deal with GOTOs and GOSUBs as well), but it also makes for hideously untidy listings as the Lynx does not justify lines for you. Imagine beginning a program with lines 100-150 in increments of 10 and then adding 120.1234, 120.1235, 130.12, 130.13, 130.44, 140.5 and 141.444 (which you'd have to be mad to do). Write it down — unjustified — if you want to see the effect it would have on readability.

Furthermore, had you been enough of a lunatic to number your lines in increments of one (or even 0.0001) and you wanted to add more lines, all you'd need to do would be to type RENUM 1,100 (ie, from line 1 at an increment of 100) to rectify the situation.

The next oddity in Lynx Basic lies in the number of loops and conditionals provided. As well as the conventional IF...THEN...ELSE, FOR...TO...STEP...NEXT, GOTO, GOSUB...RETURN, you are armed with labels (for use with GOTO and GOSUB), procedures, WHILE...WEND, REPEAT...UNTIL and three logical operators — AND, OR and NOT. This is actually a good idea: you can plough through your algorithms, relentlessly structuring as you go and then understand them afterwards.

A lot of really odd facilities are provided. These include (most notably) the ability to change the cursor character and the rate at which it flashes. The flashing effect is produced by a block character and a space being printed alternately very fast (there is no way to make characters flash from Basic apart from using this idea with PRINT), so using the CCHAR command you specify two characters which alternate. CFR (cursor flash) ranges from incredibly slow (CFR65535) to incredibly fast (CFR1).

Debugging is made easier not only by a flow trace, but also by a command, SPEED, which sets the rate of program execution. This simply increases the delay between program lines and you can then see exactly what's going on.

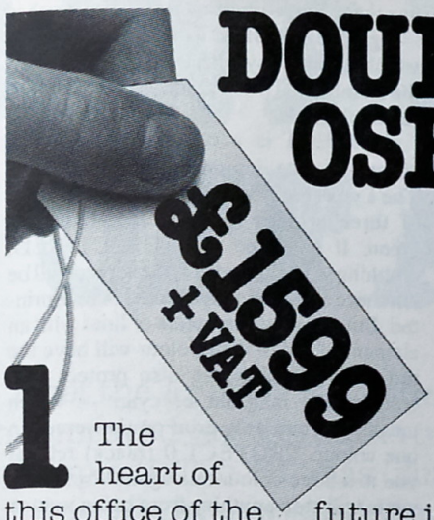
This leads us logically on to editing. I found this fairly easy, but with one or two inevitable disadvantages (the perfect screen editor has yet to be invented). Control 'E' gives you the response 'line number?' and expects either a line number, label name or procedure. Inputting one of these and pressing Return gives you the specified piece of code with the cursor at the beginning of it. You then move the cursor to the place where the correction is needed and delete/insert as necessary. The up and down arrows can move you either to the beginning or to the end of the line. Control 'Q' will allow you to edit a line, entry of which has just given an error message without specifying its number.

It would have been a little more logical to add the ability to move up and down, using the arrow keys, line by line, although the Control 'E' idea can save time in long programs. What really is a bit of a nuisance is the use of 'DEL'. You can't just type in a line number, hit Return and see it magically disappear as you can with most micros, although you can overwrite lines. You can delete blocks of lines using the format DEL first line number, second line number. The only real advantage with this method is that you have less chance of erasing lines by just typing in the numbers, although you could still overwrite lines by using the wrong numbers for new ones.

A 'shorthand' facility can be used when entering programs. It is effectively divided into two different methods. One method is referred to in the manual as 'single-key' keyword entry but is in fact double key entry. It works like this: You hold down the ESC key and type one letter standing for the reserved word you want to enter. Not all words are provided for in this manner, but all the common ones — eg, PROC, GOTO, STOP, REM, etc — are. The trouble is that you have to remember which letter is which word, although most of these are logical.

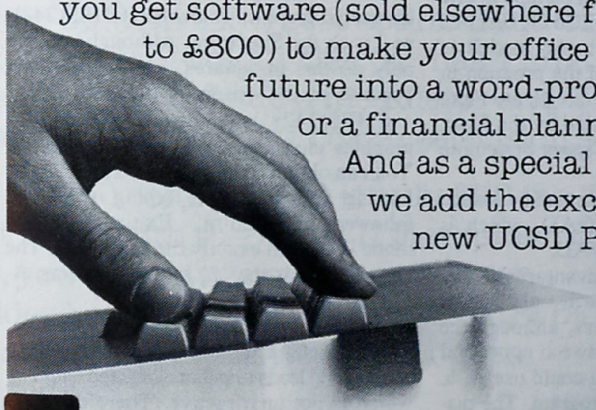
The other abbreviation facility is far

DOUBLE DENSITY OSBORNE AND PRAXIS



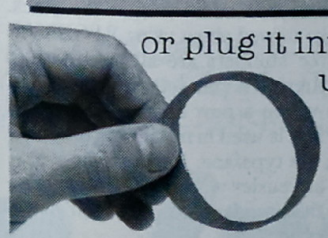
THE OFFICE OF THE FUTURE TODAY

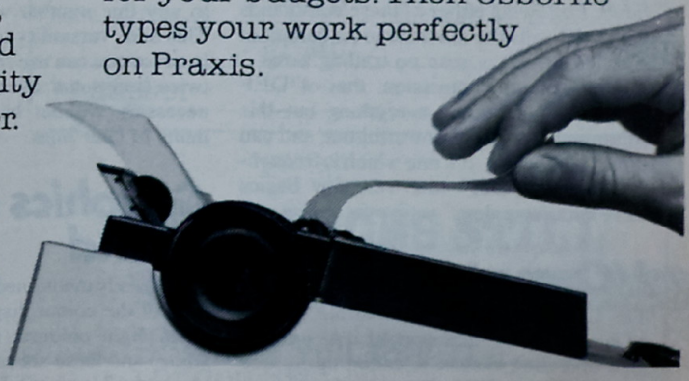
1 The heart of this office of the future is the powerful new "double density" Osborne 1 portable computer with 64k memory, two 200k disc drives and built-in monitor. The Osborne is linked to an Olivetti Praxis 30 11cps daisy-wheel printer/typewriter for printing your work superbly. In addition you get software (sold elsewhere for up to £800) to make your office of the future into a word-processor or a financial planner. And as a special bonus we add the exciting new UCSD Pascal.



2 Your Osborne and Praxis are a remarkable combination. The machines are so portable you can carry them both home instead of working late. Plug them together slip the Wordstar/Mailmerge disc into the drive and you have a word-processor with twice the capacity of earlier Osbornes. Change discs for Supercalc and you have the power of a dozen clerks for your budgets. Then Osborne types your work perfectly on Praxis.

3 The Olivetti Praxis 30 is a new style electronic typewriter with its own memory, automatic erase, and a double keyboard which gives you 14 extra characters. Use your Praxis on its own as a prestige easy-to-use typewriter, or plug it into your Osborne and use it as a high quality computer printer.





O-COMPUTERS
A member of the C/WP Group
104-106 Rochester Row, London SW1
Telephone: 01-828 9000

more reasonable. You simply type in enough letters of a reserved word (the first two are usually enough) to make it distinguishable from any other command and follow them with a full stop. Once you get used to this it is very useful. In addition, you can use '?' instead of the word PRINT.

String handling is very odd. For a start there is the glaring omission of STR$ which converts a numeric variable into a string. In spite of this, VAL is included, which does the opposite. All the old faithfuls — LEN, ASC, LEFT$, RIGHT$, MID$ and CHR$ — are there with the addition of UPC$, an oddity which will convert all letters in a string to upper case.

String arrays were, incredibly, nearly left out altogether originally. Camputers plans to introduce a string/data file handling package on either EPROM or ROM cartridge in the near future. The situation now is this: you can type DIM A$(6) which allows you to input a string of up to six characters — rather pointless in the face of such commands as INPUT A$. DIM A$(6)(12) dimensions a 12 element array, each array consisting of up to six characters. The largest eight-character/element array that the 48k Lynx will accept is 1000 elements. Only single dimensional string arrays are allowed.

While still on the subject of arrays, two-dimensional numeric arrays can be wangled with DIM A(R*C). Selection of a particular element — say 3,4 — is done using the format A(3*C+4). A bit weird really, all things considered.

A positive arsenal of numeric functions is provided. Among the more bizarre of these are INF, which returns the largest number the Lynx can process, 9.9999999E +63, and FRAC which is the direct opposite of INT. Arcsines, cosines and tangents are available (as well as SIN, COS and TAN). Two functions I've never encountered before on a standard machine are DIV — integer division (dividing without leaving a remainder) and MOD, its opposite, which gives just a remainder. Logarithms and Antilogarithms (and natural logs, too) can be used. Another Lynx oddity is the command pair ROUND and TRAIL. ROUND OFF will switch off the computer's automatic rounding facility. Internally, the Lynx works on an accuracy of eight digits and round this off to six when final values are printed on the screen. Therefore you have the choice between six and eight digit accuracy. TRAIL ON allows you to put trailing zeros on to bring accuracy up to whatever ROUND has dictated. Following either of these commands with OFF sets the machine to its default of six-digit accuracy with no trailing zeros.

Again, another omission: that of DEF FN. You can't have everything but this command is a rather powerful one and can save a lot of work. It's one which is frequently invaluable and which very few Basics leave out these days.

I/O control

The Lynx is fully disk-compatible — CP/M is about to be launched for it — and the disk operating system is called up by one command — DISK. This is, of course, a stupid thing to type in if you have no disks as it crashes the system.

Three printer commands can be used — the obvious LLIST and LPRINT; and the addition is that of LINK, which displays simultaneously on printer and screen. This is handy, can be impressive in graphics programs, and is unusual on home computers. I must point out here that I was unable to run a printer from the Lynx and can only judge these commands from the manual.

Cassette handling is good. For a start the baud rate is user-specifiable in increments of 300 from 600 to 2100 baud. This is done simply by typing TAPE followed by a number from zero to five (zero being 600 baud, the default). The manual states, however, that to use the higher baud rates you have to have a 'good quality cassette player and tapes'.

LOAD, SAVE and VERIFY are, of course, provided, with the interesting and useful addition of APPEND. This allows you to add a program stored on tape to one already in memory — provided the first line number of the program to be loaded is higher than the last one of the program in memory. Cassette recorders with a remote control facility are controlled by the Lynx — although only certain types or remote control will work.

Machine code programs must be loaded using the command MLOAD, which is otherwise exactly like LOAD.

One of the major disadvantages of the Lynx is the fact that it will only accept single-letter variable names, although the interpreter distinguishes between upper and lower case. Therefore you could use A, a, A$ and a$ in the same program. The disadvantage here does not lie in numbers as much as in distinguishing one variable from another. Very few Basic programs will need more than 52 numeric and 52 string variables. The problem really lies in being unable to read a program and understand variable names logically, so a stringent record would have to be kept of which variable stores what. Two-letter names would have been more logical, and long variable names better still.

SWAP is a really useful command which allows you to simply swap the values of two variables.

The RESTORE command can be reset to any line number which contains data. This adds versatility to DATA statements because you can use some data once, some twice and some over and over again as necessary without having to repeat data items in later lines.

Graphics and sound

I've already mentioned the surprising crispness of the colour display. The Lynx provides eight colours (including black and white) and these are coded from 0 to 7 (0 is black and 7 is white). Interestingly, you can use the colour's name as well as its number — eg, PAPER BLACK or INK GREEN which I found myself using far more than the numbers. This also improves program readability.

PROTECT is a command which can generate some interesting visual effects. The Lynx sends colour signals as a mixture of three primary colours: red, blue and green. If you type in PROTECT RED, anything in red already on the screen will be left there and cannot be cleared or overprinted. But any new characters or lines with an element of red in their colour will have the red removed. You can also protect sub-colours like magenta or cyan — which means you can only print on the screen in one colour. PROTECT 0 (black) returns you to a three-colour mix again. Protecting white will stop anything from being written to (or erased from) the screen.

This is lots of fun and it can even be unnerving if you forget that anything already on the screen in the protected colour cannot be moved. It actually looks as if the computer's crashed (hard crash I mean) if you've just listed a program and then run it with a PROTECT statement before a CLS statement.

Colour mixing is possible using a fine chequered graphics character (CHR$(242)) as it is on the Spectrum. Although you can only do this in squares, it's very striking because of the crispness of colour.

CHR$(242) is one of 26 predefined graphics characters listed in the manual. You can obtain these from the keyboard by locking the shift and typing CTRL 1 followed by 'Return'. Exiting graphics mode is done in exactly the same way. The keys used are the '£' key and letters A-Y.

You can also define your own characters. The Lynx character matrix is 6x10 which is unusual and leads to rather elongated lettering, which is not unattractive. Therefore each character takes up 10 bytes of memory and this has to be reserved before you start inputting the data. I'll deal with that in more detail later, along with machine code.

Once you've reserved your memory, the binary digits to make up the character's bit pattern can be read from DATA statements using the command BIN — not *a la* Spectrum but to the format POKE address, BIN(A) after the data has been read. The Lynx has two pointers and one read-only variable in memory, ALPHA — the beginning of the conventional character set, GRAPHIC — the beginning of the duplicate character set and HIMEM which tells you the first free address after the stack. This can be moved about using RESERVE. These three numbers are all important in defining your own characters. GRAPHIC points to where a new character is stored and ALPHA is used in modifying letters — changing the typeface, for example.

There are easier ways to design characters, but this is adequate once you've got the hang of it.

The Lynx's screen resolution is 256x248. This allows for some impressive drawing effects. DOT, DRAW, MOVE and PLOT are the relevant commands: DOT does the obvious thing — puts a dot on the screen at the specified coordinates; DRAW puts a

line on the screen from the last point plotted (or the last position of the graphic cursor) to the point specified; MOVE simply moves the invisible graphic cursor to where you tell it to go. PLOT struck me as being rather unnecessary because it has five modes combining all the other graphics commands along with a relative move and draw, ie, the coordinates represent the amount by which the cursor moves, not the position at which it ends up.

Line drawing is crisp, clear and fast. Lines can be put on the screen at quite steep diagonals without an ugly 'stepping' effect becoming too apparent. My big reservation about Lynx graphics is that there is no easy way to fill a shape. The only way I could find to do this was by plotting individual dots, which is slow and inaccurate. It may not have been possible, but a PAINT or FILL command would have made a good replacement for the rather obsolete PLOT (or PLOT alone could have been used and all the other words scrapped, perhaps). Pixels are individually colour-definable and, when the DOT command is used, they prove to be very fine although they are clear in any colour.

An interesting idea is the use of PRINT @ and WINDOW. PRINT @ uses columns two pixels wide (as does WINDOW) and rows to the highest resolution of 248 to put letters exactly where you want them on the screen. This is far more versatile than using character squares. You can therefore mix text and graphics tidily if you want to. WINDOW specifies the size of the text area. You specify the first column and the last column + 1 and the first row and the last row + 1 to change the part of the screen on which you want to print. Some fascinating effects can be achieved here by using WINDOW in conjunction with PROTECT. Again, there is one reservation in my attitude to this command: some kind of quick normalising command would have been handy to reset the window to its default. As it stands, you have to remember the coordinates of the normal text window and use those numbers to put things back to normal.

Another Lynx oddity is VDU. This uses CHR$ (ASCII) codes 0 to 31 to perform cursor movements and other graphics commands as well as to perform the power-up beeping noise. You can use PRINT CHR$ to do the same thing, so I found VDU rather unnecessary, although it is quicker to enter than PRINT CHR$. Three of the facilities here are really striking. The first of these is VDU 21 — overwrite. This is like the Spectrum's OVER command and is turned off by inputting VDU 20. Secondly, there's VDU 24 — double-height characters (turned off by VDU 25). These are really fun to use and striking effects can be obtained by mixing single and double height text. Lastly, and more subtly, you can employ superscript and subscript on text. This simply means moving the cursor up or down three pixel lines, giving a peculiar overlapping effect. This command has to be repeated for each line of super/subscripted text. The other VDU codes do things like carriage return, colour changes, line feeds and backspace.

Sound is based on a digital/analogue conversion principle and is used with two commands, BEEP and SOUND. BEEP is followed by three figures broken up by commas; these stand for wavelength, number of cycles and volume. Volume is specifiable from 0 to 63 and wavelength and number of cycles from 0 to 65535. High wavelengths are high-pitched sounds (for those of you who didn't do physics at school), so to make a high note of the same duration as a low one, the result of multiplying the number of cycles by wavelength must be the same. The large number of parameters available makes for an impressive range of sounds.

SOUND is a machine code based command. You can POKE several consecutive addresses with appropriate values, type in SOUND followed by the first of these addresses and the computer will read them, converting the values within to sounds as it goes, stopping when it reaches a value of 0. This could conceivably be used to synthesise speech.

Machine code is well provided for, with DPOKE and DPEEK complimenting POKE and PEEK by enabling you to read or write to two consecutive addresses with one word. Using both you specify the first address. DPOKE loads the least significant byte into the address and the most significant byte into the address+1.

PRINT# is a decimal-hex converter. It works to four digits and the largest decimal number which can be converted is 9,999,999. Only the integer part of the number is used. Hex can be input directly by prefixing the Hex number with '&'.

Binary And, Or and Exclusive Or are provided and a command, CODE, allows you to incorporate machine code into Basic programs by prefixing each line with CODE. INP and OUT read or send data from/to specified Z80 ports.

A machine code monitor is accessible from Basic by typing MON (and exited by typing 'J'). This includes a breakpoint facility as a debugging aid.

The Benchmark timings tell a lot, in this case, about the machine. The first two are fairly fast but BMs 7 and 8 are really slow. BM8 involves finding functions and the Lynx obviously does this rather too slowly. It's turned out slower overall than many other Z80 machines and is certainly in the lower end of the overall timings range.

To conclude on the subject of Lynx Basic, I've really got no choice but to express stern reservations. Some of the effects which can be obtained with ease are really striking, but the overall structure is not in keeping with the reasoning behind it. At first sight it looks fantastic from a programming point of view, but using it for a while reveals serious disadvantages.

To a naive user, in any case, the importance rests far more on things like the efficiency of the screen editor than the intricate sophistication of the Basic. I'll make a huge criticism here while I'm at it: the screen does not scroll. It goes up to the top line and overwrites what was there before. This makes for either a hideously untidy screen or for the hassle of using CLS every time you want to list a program. And if your program is more than one screenful you've got real readability problems.

Lynx Basic has several omissions which appear to have been replaced by unnecessary trimmings. It repeats itself — as in the use of VDU and CHR$ — thereby leaving out more important things. The lack of a reset key makes it possible to get into a terminal crash very easily — for instance, if you get stuck in an input loop you can't use ESCape to get out of it (it has no effect, more to the point) and you have therefore to switch the machine off. A little thought here would have made all the difference.

Benchmark timings

BM1	1.7
BM2	4.3
BM3	12.4
BM4	8.9
BM5	10.4
BM6	16.3
BM7	29.9
BM8	86.6
Average	34.1

For a full explanation of Benchmark timings, see PCW, Nov 82.

Documentation

A paperback user manual comes with the machine, written in the familiar home micro 'step-by-step' style. It's illustrated with bizarre pictures of little robots wearing Elizabethan-style breeches which drew chuckles of not-too-complimentary amusement from me as I read it.

I'm not too happy about it, either. It starts on a good note, takes you through the fine basics in a readable manner and is written in a friendly style. It has some attractive demonstration programs in it but it does not go into enough depth.

Some explanations are sadly lacking — for instance that of user-defined characters — and others still have mistakes in them. I understand, however, that the manual will be amended.

Furthermore, no useful memory addresses are provided — for instance, video RAM start and end addresses — which would make the enthusiast's endless pursuit of machine gymnastics a great deal easier. As with many machines a manual twice as long would have been a great deal more helpful and the addition of more involved technical information a blessing. But other machines are guilty of the same fault and I must confess to being rather a documentation fundamentalist, especially where micros for beginners are concerned. Ideally you shouldn't, in these cases, have to buy other books to explain the bits your computer leaves out. With the Lynx (among others), you do.

Software

There's not a lot to be said about this outside of speculation. Camputers tells me the machine is aimed at the 'Sinclair graduate'.

GOTO page 158

GOODBYE TO FLOPPIES C/WP INTRODUCES A HARD DISC FOR £995.

C/WP have done it again. We proudly announce the C/WP CONTOUR, a range of British-made high technology 5-inch Winchester discs at prices starting below £1000.

Suddenly your micro-computer has come of age. One little box, not much larger than a floppy disc drive, stores up to 21 million characters. That size costs £1995 and stores the equivalent of more than 160 floppy discs, or 35 full-length books. And in a maximum of around a fifth of a second the C/WP CONTOUR finds any piece of information your computer asks for.

C/WP CONTOUR is available now for your APPLE II, IBM Personal, Sirius, Superbrain and many other CP/M and S100 machines. On APPLE the C/WP CONTOUR supports DOS 3.3, Pascal and CP/M operating systems.

Write or telephone for full details of this great price breakthrough. Now you can afford the luxury of a Winchester.

C/WP CONTOUR PRICE LIST	
3Mb (formatted)	£995
6Mb (formatted)	£1195
12Mb (formatted)	£1595
21Mb (formatted)	£1995
Tape streamer back-up (21Mb 4min)	£1495

All prices exclude VAT

C/WP FOR BIG VALUE IN HARD DISCS

C/WP

C/WP Computers
108 Rochester Row, London SW1 1JP
Telephone 01-828 9000

THE LYNX

Continued from page 142

This should therefore mean that sophisticated games and home software are published for it. At present three sets of games are on their way — and these will probably speak for themselves if you read a few adverts. There are also plans for educational software. Camputers sees the Lynx as having a future in schools. There is, though, no software currently available for the machine apart from a demonstration tape given with the machine when it is purchased.

It's claimed that disk drives will be launched for the Lynx in March/April. If the Lynx itself is any guide, this could turn into August. The machine is CP/M compatible and this, coupled with upper-range memory expansion and therefore 80-column display, could make the Lynx into a true dual-personality machine. Camputers is working on a manufacturer agreement with a software house (I'm not told which one) to put word processing and accounting software on the Lynx. Pascal, Forth and Comal compilers will also arrive, initially on cassette and then on disk.

EPROM additions and ROM cartridges are planned for the near future. One of these is a data file handling package which will also be available on cassette. A full graphics/sound add-on is also on the cards.

The Lynx is a very 'soft' machine and many of its capabilities depend on the quality of software produced for it. Its impact on the market also depends on its resident software — ie, Basic — which does leave more than a little to be desired. It will probably happen that other Basics (and other languages as well) are written to replace what is there already if people feel a demand for it.

Expansion

I've already mentioned the disks. These, says Camputers, will be single-sided, double-density with a capacity of 250k per drive. There's the option of either single or twin disk units and a disk unit includes a Centronics interface. As well as CP/M, a 'primitive' Lynx DOS has been pointed at for the future but it's probably better to go for CP/M when it's available (which will be at the same time as the disks).

Memory expansion is great on the face of it but it's essential to remember that if you buy a 192k machine you only have 64k of RAM immediately available for programming. The rest is video/graphics RAM but this can, apparently, be accessed using machine code. Details on how this is done will be made available to users.

The Lynx printer will be a tiny printer of the 'till roll' type and will cost in the region of £60. For more serious minded people, the recommended printer, the Seikosha 250X, can be run from a Lynx through the RS232 using a special lead and software available from Camputers. Prices of these are not available yet.

A light pen is also on its way but Camputers will not commit itself to a date on that. It will be connected to the Lynx through the port marked 'LP' which also runs a black & white monitor. An RGB port is provided for a colour monitor.

A look at the prices listed will reveal that you can set yourself up with a theoretically decent business machine for £299 (96k Lynx with 64k user RAM) + approx £400 (twin disks) + approx £100 (black & white monitor) + approx £250 (Seikosha printer) which comes to about £950. Then on top of that you've got to buy software. Better deals have been known. Unless you've had a Lynx for a while and don't want to sell it this can't be said to be the best proposition if you only want to do stock control or whatever.

Support and availability

Camputers has made dealer agreements with Spectrum Computer Centres and Lasky's who should have the Lynx in stock by the time this Benchtest is published (and don't write to me if they haven't. . .). No plans for further dealers are in the air as Camputers doesn't want to encounter stock flow problems. It plans to have 40,000 Lynxes produced by the end of this year.

A 12 month warranty is wisely tacked onto the whole package but this excludes labour. Servicing after that will be carried out by the subcontractors and by 'one of the retailers'. Just which one will soon become apparent.

A Lynx newsletter will be published and sent out to every user who sends back the guarantee card.

Technical specifications

CPU:	4 mHz Z80A
RAM:	48k (16k graphics, 16k user) expandable to 192k total
ROM:	16k
I/O Ports:	Expansion bus, cassette, RS232, RGB, Composite video
Disks:	Single or Twin, SS/DD, 250k/drive
O/S:	CP/M-80
Languages:	Basic. Pascal and Forth to come.

Conclusions

I'll begin by saying that Camputers has, as a company, the best intentions and follow that up with 'the best-laid plans of mice and men. . .'. You see, there have been problems. So far the machine itself has been late and there are no definite dates on other hardware to come. If you bought one now

GOTO page 223